A Co-cooperative Evolutionary Algorithm for Flexible Scheduling Problem under Uncertainty

Wang, Yan; Lin, Lin; Gen, Mitsuo; Sun, Lu; Kawakami, Hiroshi

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 61 (2015) 515 – 520
Procedia Computer Science

Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

A Co-cooperative Evolutionary Algorithm for Flexible Scheduling Problem under Uncertainty

Yan Wang[a], Lin Lin[a,b,*], Mitsuo Gen[b,c], Lu Sun[a] and Hiroshi Kawakami[d]

[a]*Dalian University of Technology,* 116620*, China*
[b]*Fuzzy Logic Systems Institute,* 820-0067*, Japan*
[c]*Tokyo University of Science,* 113-8656*, Japan*
[d]*Kyoto University,* 606-8306*, Japan*

Abstract

Flexible Manufacturing System (FMS) has the characteristics of resources non-uniqueness; the operation can be performed by any available machine in a set of machines. Due to reduce the constraints of the machine, it becomes higher flexible. But it has high complexity existing in the actual production system and making its complexity is higher. We experiment three classical evolution algorithms and two modified evolutionary algorithms with grouping mechanism, and do the experiments under the certain environment on different size of data. We found that as the data growing, the evolutionary algorithms with grouping mechanism can get a better solution with larger probability. In this paper, we propose hybrid evolutionary algorithm based on the particle swarm algorithm combining a set-based grouping and parameter adaptive adjustment mechanism. It is given a set number of available groupings, choose a grouping number and calculate adaptive value, if adaptive value becomes better. Through experiments, we conclude that the proposed hybrid evolutionary algorithm based on co-cooperation gets better solution than evolutionary algorithm and then improve robustness of the proposed algorithm.

Keywords: Co-cooperation Evolutionary Algorithm, Flexible Job-shop Scheduling Problem (fJSP)

1. Introduction

Since the 21st century, due to the fierce competition and complex technology, small batch of modern manufacturing mode has become the development direction of manufacturing enterprise production and business operation mode under the background of demand diversity. In order to further improve the production efficiency and flexibility of production and reduce the cost of equipment, a flexible job shop scheduling problem (fJSP) model with numerical control (NC) technology replaces the traditional scheduling model. The fJSP as an extension of the job

doi:10.1016/j.procs.2015.09.205

shop scheduling problem (JSP), breaks through the constraints of resources uniqueness: each operation can be processed on every available machine in the machine set, but the processing time of each operation on every time is fixed, so that fJSP can improve the production efficiency, shorten the ordering cycle and increase the rate of orders delivered on time. Gao, *et al* developed a new hybrid genetic algorithm (HGA) to solve the fJSP models with non-fixed availability constraints (Gao, Gen & Sun, 2006; Gen, Cheng & Lin, 2008) and also proposed hybrid genetic and variable neighborhood descent algorithm for solving fJSP models (Gao, Sun & Gen, 2008). Gen, *et al* proposed a multistage-based GA with bottleneck shifting developed for the fJSP model (Gen, Lin & Gao, 2009).

However, in actual production process, the processing time are often changed due to the increasing or decreasing of operations, the machine becomes from available to unavailable and so on. In other words, the processing time of each operation is not fixed but stochastic. So, stochastic flexible job shop scheduling (S-fJSP) model is closer to the modelling of scheduling problem in actual production systems.

Recently, Horng *et al* proposed an evolutionary algorithm of embedding evolution strategy (ES) in ordinal optimization (OO), abbreviated as ESOO, to solve for a good enough schedule of stochastic job shop scheduling problem (S-JSP) with the objective of minimizing the expected sum of storage expenses and tardiness penalties using limited computation time (Horng, Lin & Yang, 2012). They embedded the ES into sequence evolution (SE) in order to minimize the cost of storage and punished for being late which is aimed on the sum of progress and expectations. Wang *et al* proposed an effective estimation of distribution algorithm (EDA), so that, new individuals can be generated among the search region with promising solutions by updating the probability with a mechanism and sampling by the probability model (Wang, Wang, Xu & Liu, 2013). Lei proposed an efficient decomposition-integration genetic algorithm (DIGA) and a co-evolutionary genetic algorithm (CGA) to minimize the maximum completion time (Lei, 2012). DIGA used a two-stage representation, an efficient decoding method and a population to increase the best solution; CGA used chromosome of a novel representation consists of ordered operation list and machine assignment (Lei, 2010; Lei, 2012). Niu *et al* proposed an assignment-first decomposition (AFD) and a sequencing-first decomposition (SFD) for solving the problem (Niu, Sun, Lafon & Zhang, 2012). Hao, *et al* recently proposed a cooperative EDA for semiconductor final test scheduling problems (Hao, Wu, Chien & Gen, 2014).

However, the methods mentioned above are all overall do the evolution operations, with the increase of the scale of problems, the effectiveness of the algorithm is limited even goes down. This is a serious problem in real production system that the scale of problem is often large. So it is lack of related research for flexibility and the effectiveness analysis of the scheduling for designing optimization method, especially for using the framework of cooperative coevolution and grouping the variables to increase the effectiveness depending on the problem scales.

2. Scheduling Model under Uncertainty

In S-fJSP, each job i consists of n_i operations (O_{i1},O_{i2},...,O_{ini}). For each operation O_{ik}, processing machine must be from the machine set A_{ik}. The difference between S-fJSP and fJSP is that processing time is randomly given by expectation $\mathrm{E}\left[\xi t_{ij}\right]$ and variance v_{ij}, such as normal distribution, joint distribution and random distribution. In order to test the robustness for each solution, we randomly assign the processing time according to the distribution for each operation.

The symbols used in S-fJSP are defined as follows:

Indices:

i, h: job index, i, h=1,2.,…,n

j: machine index, j=1,2,...,m

k, g: operation index, k, g=1,2,…, n_i

Parameters:

n: total number of jobs

m: total number of machines

n_i: total number of operations in job i

ξt_{ikj} : processing time on the machine j of the kth operation O_{ik} of job i, a stochastic variable

ξC_i : processing time of job i, a stochastic variable

Decision variables:

ξc_{ik} : completed time of O_{ik}, a stochastic variable

x_{ikj}: machine j is selected for O_{ik}

The objective function (1) is to minimize the stochastic makespan and the mathematical programming model of the fJSP under uncertainty formulated as follows:

$$\min \quad E[\xi C_M] = E[\max_{i=1,\ldots,n}\{\max_{k=1,\ldots,n_i}\{\max_{j=1,\ldots,m}\xi c_{ikj}\}\}] \tag{1}$$

$$\text{s. t.} \quad \xi c_{ik} \quad \xi c_{i(k-1)} \geq \xi t_{ikj} x_{ikj}, k = 2,\ldots,n_i, \forall i, j \tag{2}$$

$$[(\xi c_{hg} \quad \xi c_{ik} \quad \xi t_{hgj}) x_{hgj} \geq 0] \vee [(\xi c_{ik} \quad \xi c_{hg} \quad \xi t_{ikj}) x_{ikj} \geq 0], \forall i, j, g, h \tag{3}$$

$$\sum_{x_{ikj} \in A_{ik}} x_{ikj} = 1, \quad \forall i, k, j \tag{4}$$

$$\xi c_{ik} \geq 0, \quad \forall i, k \tag{5}$$

$$x_{ikj} \in \{0,1\}, \quad \forall i, k, j \tag{6}$$

The constraints (2) - (3) represent the operating sequence and avoiding the duplicated machine assignment constraints, respectively. The constraint (4) guarantees machine allocation that for each operation can assign on one machine from machine set at one time. The constraints (5) – (6) are nonnegative and 0 – 1 binary variable restriction on decision variables, respectively.

3. Proposed algorithm CChEA

3.1 The Co-Cooperative hybrid Evolutionary Algorithm (CChEA)

JSP is a typical combinatorial optimization problem and it is a NP-hard problem under the constraint of priorities and resources (Lawler, 1993; Kennedy, 1995). So we choose the heuristic algorithm to solve S-fJSP. Evolution algorithm (EA) is inspired by natural selection, it makes the individuals with higher ability of survival be reserved and the gens with higher adaptability be spread to more individuals, so the species of evolution will be more and better to adapt to their environment by the methods mentioned above.

```
procedure: Co-cooperative Hybrid Evolution Algorithm
input: data set; parameters
output: best solution of S-fJSP
begin
    t ← 0;
    initialize population p(t) by random value;
    get sub-p(t) by random grouping;
    g(t) ← 0;
    while( not meeting termination condition)
        evaluate p(t) and calculate the best fitness;
        if (gbest(t) has not become better )
            choose s from S randomly;
            reconstruct sub-swarms for all n dims;
        for each swarm
            if (t meats the condition)
                adapt the parameters by formulas;
        record correlation parameters;
        t ← t+1;
    end
    output best solution gbest(t) of S-fJSP
end;
```

Fig.1. Pseudo-code for Hybrid Evolution Algorithm

Genetic algorithm (GA) is a typical classical EA, it presents the solutions of problems by chromosomes, uses the generation operations to generate new individuals and gets the best solution by iteration. Particle swarm optimization (PSO) is another typical EA. It simulates the process of the birds' predation, changes the moving speed and position according to the formula to reach all parts of the searching space. According to the character of the encoding method in the PSO, it can has larger searching space than GA (Kennedy& Eberhart, 1995). So we use PSO as the basic algorithm of the proposed algorithm. Then, we use the set-based random grouping mechanism with the set-based grouping and parameter adaptive adjustment mechanism. It is given a set number of available groupings, chooses a grouping number and calculates adaptive value. If adaptive value become better, we will use the grouping number continually; otherwise, when the first time that good adaptive value does not change even worse, we abandon the grouping number, randomly choose another number in the grouping set. We named it as co-cooperative hybrid evolutionary algorithm (CChEA). The pseudo code of CCHA is shown in Figure 1.

PSO uses the follow formulas to update velocity and position in each generation t, respectively. The initial values are random. Parameter ω presents inertia weight, $rand_1$ and $rand_2$ are random value within [0,1]. The bigger ω presents affecting by chromosome's own value; the smaller ω presents affecting by population's social factor.

$$v_i(t+1) = \omega\, v_i(t) + c_1 rand_1[p_{pbest}(t) - x_i(t)] + c_2 rand_2[p_{gbest}(t) - x_i(t)] \quad (7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

3.2 The Set-based Random Grouping Mechanism

The grouping method is inspired by the divide-and-conquer algorithm: to solve the problem is to change the structure of the organization dynamically, we called this as random grouping method. Decompose the problems of N dims into k sub-problems (each sub-problem has $s=N/k$ dims) and the sub-problems do not effect each other. But they all belong to the same optimization process. The correlation between two random selected variables in one sub-problem will increase with the increasing of the number of iterations. N=1000, k=10, $s=N/k$=1000/10=100, the number of generation is 50, the times of random grouping is also 50, we present the mathematical proof as follow:

$$p(x \geq 1) = p(1) + p(2) + \ldots + p(50) = 1 - p(0) = 1 - \binom{50}{0}(0.1)^0(1-0.1)^{50} = 0.9948 \quad (9)$$

Among them, $\boldsymbol{x}$ presents 2 variables were assigned in the same group, $p(m)$ presents m times. Due to the dims in one sub-problem will influence directly, so we give a set S={2, 5, 50, 100} to adapt the grouping size dynamically in the repeated iteration. If the fitness becomes better, we keep current grouping size; otherwise, when the first time that good adaptive value does not change even worse, we randomly select a number from the set and replace the current grouping size.

3.3 The Parameter Adaptive Adjustment

Due to the existing experiment parameters have important influence on the results and decide the evolution operations, so the inertia weight and random value are particularly important to the result. So if they can be adapted according the merits of the fitness value in the process of experiment, there will be a good influence on the performance of the algorithm.

The random value $rand_1$ and $rand_2$ of the PSO update formula determines the influence of the two parts, they are initialed by random values. In the process of experiment, they are confirmed to the normal distribution with average value CRm and the standard deviation 0.1.

$$rand_{1or2} = N_i(CR\text{m}, 0.1) \quad (10)$$

CRm is firstly set as 0.5. These CR values for all individuals remain the same for 5 generations and then a new set of CR values is generated using the same equation. During every generation, the CR values associated with offspring successfully entering the next generation are recorded in an array CRrec. After 25 generations CRm will be updated:

$$CR\mathrm{m} = \frac{1}{|CR_{\mathrm{rec}}|}\sum_{k=1}^{|CR_{\mathrm{rec}}|} CR_{\mathrm{rec}}(k) \tag{11}$$

The inertia weight ω is adapted by formula (12), we first set the selectance *fp* as 0.5, if the random value is smaller than *fp*, ω is confirmed to the normal distribution with average value *0.5* and the standard deviation 0.3, and otherwise, ω is confirmed to Cauchy distribution with parameter 1. The inertia weight ω is adapted each 15 generations.

$$w_i = \begin{Bmatrix} N_i(0.5,0.3), & \text{if } U_i(0,1) < fp \\ \xi_i, & \text{otherwise} \end{Bmatrix} \tag{12}$$

The selectance *fp* is adapted by formula (13), among them, After evaluation of all offspring, the number of offspring successfully entering the next generation while generated by normal distribution and Cauchy distribution are recorded as ns1 and ns2, respectively, and the numbers of offspring discarded while generated by normal distribution and Cauchy distribution are recorded as nf1 and nf2. Those two pairs of numbers are accumulated within a specified 50 generations, called the "learning period". Then, the probability *p* is updated as:

$$p = \frac{ns_1(ns_2 + nf_2)}{ns_2(ns_1 + nf_1) + ns_1(ns_2 + nf_2)} \tag{13}$$

4. Numerical Experiment

In order to verify the effectiveness of the proposed algorithm, Co-cooperative hybrid Evolutionary Algorithm (CChEA) in this paper, we do experiment with 4 scales of data under uncertain environment. Compared with a classic genetic algorithm (GA), a binary genetic algorithm (Binary GA), particle swarm optimization (PSO), differential evolution (DE) algorithm, cooperative coevolution group with PSO (CCPSO), particle swarm optimization with adaptive grouping differential evolution algorithm (PSO+DE), and proposed CchEA. The scales of data are 5*5, 10*10, 15*15 and 20*20. To ensure the reliability of the experimental, the experiment repeats 30 times and gets the mean value. Test machine is Intel(R) Core(TM) i3-2120 CPU @3.3GHZ, 4GB.

Table. 1 Experimental parameters Settings

	5*5	10*10	15*15	20*20
Pop. size	10	10	50	100
Crossover prob.	0.5	0.5	0.5	0.5
Mutation prob.	0.5	0.5	0.5	0.5
Terminating condition	Num. of Evolved Individual=5000	Num. of Evolved Individual=5000	Num. of Evolved Individual=5000	Num. of Evolved Individual=10000
ω	0.5	0.5	0.5	0.5
c	1	1	1	1

Table. 2 Experimental results of 5*5 – 20*20 under uncertain

		GA	BinaryGA	DE	PSO	PSO+DE	CCPSO	CChEA
5*5	mean	403.780	856.111	449.400	444.070	442.265	357.640	323.880
	variance	2475.886	5842.077	4124.570	2567.000	6241.600	4158.915	2165.183
10*10	mean	1084.630	3054.621	1145.137	1654.510	1081.229	773.300	752.810
	variance	40970.330	37297.220	44636.180	19871.420	23810.000	11920.960	14777.628
15*15	mean	1636.421	6575.229	1601.387	1654.511	1707.730	1351.139	1294.890
	variance	33869.900	51743.600	17060.140	19871.421	27839.430	26135.642	17024.820
20*20	mean	2289.632	9468.150	2279.660	2326.364	2290.110	1890.340	1780.440
	variance	75840.620	65489.100	41608.780	55377.230	43570.890	35041.120	28763.126

Table 2 shows the experimental results of 4 scales of problems under uncertain, we used a different color to label each the best of each attribute, and we could found that CChEA has better performance for different scales of problems and compared to the algorithms in references (Della *et al*, 1995, Sinha *et al*, 2003, Kenddy *et al*, 1995, Li *et al*, 2012; Yang *et al*, 2008).

5. Conclusion

We proposed Co-cooperative hybrid Evolutionary Algorithm (CChEA) for solving flexible Job-shop Scheduling Problem (fJSP) under uncertain environment in Flexible Manufacturing System (FMS). We used the set-based random grouping mechanism with the set-based grouping and parameter adaptive adjustment mechanism. It is given a set number of available groupings with a grouping number and calculates adaptive value. If the adaptive value becomes better, we will use the grouping number continually, otherwise randomly select a grouping number of the set. The proposed algorithm CChEA can get better solutions and increase the robustness. Meanwhile, in our future work, experiments will be processed by group under the distributed environment for each sub-population of the different and parallel processing, so that not only optimize the optimal solution, but also shorten the time more efficiency.

Acknowledgments

This work is partly supported by the Fundamental Research Funds for the Central Universities No. DUT15QY10 and the Grant-in-Aid for Scientific Research (C) of Japan Society of Promotion of Science (JSPS) No. 15K00357.

References

[1] J. Gao, M. Gen, and L. Sun (2006). Scheduling Jobs and Maintenances in Flexible Job Shop with a Hybrid Genetic Algorithm, *Journal of Intelligent Manufacturing* 17: 493–507.
[2] J. Gao, L. Sun & M. Gen (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, 35(9), 2892–2907.
[3] M. Gen, R. Cheng and L. Lin (2008). *Network Models and Optimization*: *Multiple Objective Genetic Algorithm*, 710pp, Springer, London.
[4] M. Gen, L. Lin, and J. Gao (2009). Multistage-based Genetic Algorithm for Flexible Job-shop Scheduling Problem. *Intelligent and Evolutionary Systems*, 187: 183–196.
[5] S.-C. Horng, S.-S. Lin, F.-Y. Yang (2012). Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Systems with Applications*, 39: 3603-3610.
[6] S.Y. Wang, L. Wang, Y. Xu, M. Liu (2013). An effective estimation of distribution algorithm for the flexible job shop scheduling problem with fuzzy processing time, *Inter. J. of Production Research*, 51: 3778–3793.
[7] D. Lei (2010) A genetic algorithm for flexible job shop scheduling with fuzzy processing time, *Inter. J. of Production Res.*, 48:2995–3013.
[8] D. Lei (2012). Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling, *Applied Soft Computing*, 12: 2237–2245.
[9] G. Niu, S. Sun, P. Lafon and Y. Zhang (2012). Two decompositions for the bicriteria job-shop scheduling problem with discretely controllable processing times. *Inter. J. of Production Research*, 50(24): 7415-7427.
[10] X.C. Hao, H. W. Lin, X.L. Chen & T. Murata (2012). Cooperative Bayesian optimization algorithm: a novel approach to multiple resources scheduling problem, *IEEJ Trans. on Electronics, Information and Systems*, 132(12): 2007-2018.
[11] E.L. Lawler, J.K. Lenstra, A.R. Kan, and D.B. Shmoys (1993). Sequencing and scheduling: Algorithms and complexity, Chapter 9, *Handbooks of Operations Res. & Management Science,* Elsevier, 445-521.
[12] J. Kennedy and R. C. Eberhart (1995). Particle swarm optimization. *Proceedings of IEEE Inter. Conf. on Neural Networks*, IV: 1942-1948.
[13] F. Della Croce, Tadei and R. Volta (1995). A genetic algorithm for the job shop problem, *Computers & Operations Research*, 22(1): 15-24.
[14] N. Sinha, R. Chakrabarti and P.K. Chattopadhyay (2003). Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on Evolutionary Computation*, 7(1): 83-94.
[15] J. Kenndy and R.C. Eberhart (1995). Particle swarm optimization, *Proc. of IEEE Inter. Conference on Neural Networks*, 4: 1942-1948.
[16] K. Price, R.M. Storn and J.A. Lampinen. (2006). *Differential evolution*: *a practical approach to global optimization*. Springer Science & Business Media.
[17] X. Li and X. Yao. (2012). Cooperatively coevolving particle swarms for large scale optimization, *IEEE Transactions on Evolutionary Computation*, 16(2): 210-224.
[18] Z. Yang, K. Tang and X. Yan. (2008). Self-adaptive differential evolution with neighborhood search, *Proc. of Congress on Evolutionary Computation IEEE.* 1110-1116.
[19] X-C Hao, J-Z Wu, C-F Chien and M. Gen (2014). The cooperative estimation of distribution algorithm: A novel approach for semiconductor final test scheduling problems, *J. of Intelligent Manufacturing*, 25(5): 867-879.